UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/748,029	12/22/2000	Ariel Cohen	00-177	4217

24319 7590 01/12/2005

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106
MILPITAS, CA 95035

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/12/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/748,029	COHEN ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- Th MAILING DATE of this communication appears on th cov r sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 October 2004 and 01 November 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-23 have been considered. Claims 1, 7, 8, 17, 18, and 20-23 have been amended as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: After Final Amendment as filed 07 October 2004 and RCE as filed 01 November 2004.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-7, 9-10, 16, 18-19 and 23 are rejected under 35 U.S.C. 102(e) as being anticipated by Patel et al., U.S. Patent No. 6,332,215.

5. Regarding claim 1, Patel has taught an apparatus comprising:

a. A processor (20 of Fig.1):

   i. Comprising a number of internal registers (44 of Fig.3),

   ii. Configured to manipulate contents of said internal registers in response to instruction codes of a first instruction set (see Col.4 lines 6-22),

b. A processor interface circuit coupled to said processor (28 and 25 of Fig. 3);

c. A memory interface circuit coupled to a memory device (24 of Fig. 3);

d. An extension stack coupled between said processor interface and said memory interface (50 of Fig.3),

e. A translator circuit (see Fig.3)

i. Coupled between said processor interface and said memory interface (42 of Fig. 3)

ii. Configured to implement a stack using one or more of the internal registers of said processor and said extension stack (see Col.4 lines 9-22). Here, the JVM stack is implemented using both the internal registers (44 of Fig.3) and the extension stack (50 of Fig.3), as not only are the java registers in the internal registers stored within the extension stack (see Col.4 lines 19-22), but the internal registers contain the pointers to the extension stack, allowing data to be written to and read from the stack (see Col.4 lines 12-17).

6. Regarding claim 2, Patel has taught the apparatus according to claim 1, wherein said one or more internal registers are used to store a top of said stack (see Col.4 lines 8-17).

7. Regarding claim 3, Patel has taught the apparatus according to claim 2, wherein said top of said stack comprises a Java virtual machine (JVM) top of stack (TOS) (see Col.4 lines 8-17).

8. Regarding claim 4, Patel has taught the apparatus according to claim 1, wherein said internal registers are dynamically allocated in response to stack status (see Col.5 line 51 – Col.6 line 11).

9. Regarding claim 5, Patel has taught the apparatus according to claim 1, wherein said translator circuit is further configured to generate one or more instruction codes of the first

instruction set for controlling the internal registers in response to an instruction code of a second instruction set (see Col.4 lines 6-22 and Col.5 lines 48-60). Here, an instruction of the second instruction set (iadd) is translated into an instruction of the first instruction set (ADD R1, R2), and the ADD instruction controls and manipulates the internal registers by reading them, performing an operation on them, and then writing back a result to one of the registers, wherein all of the registers involved are on the operand stack (see Col.5 lines 57-60), which is implemented using the internal registers (44 of Fig.3) and the extension stack (50 of Fig.3) (see Col.4 lines 17-22).

10. Regarding claim 6, Patel has taught the apparatus according to claim 5, wherein said instruction code of said second instruction set comprises a stack instruction (see Col.2 lines 19-27). It is well known in the art that Java Virtual Machine instructions are stack-based, with their operands being provided on a stack rather than in registers. It is therefore inherent that any non-trivial instructions of the stack-based Java instruction set (first instruction set), such as the "iadd" instruction, will be "stack operations" due to their stack-based nature.

11. Regarding claim 7, Patel has taught the apparatus according to claim 1, wherein said translator circuit comprises a stack management unit (64 of Fig.3) coupled to said processor interface, said memory interface, and said extension stack (see Col.5 lines 6-11).

12. Regarding claim 9, Patel has taught the apparatus according to claim 1, wherein said extension stack (50 of Fig.3) is implemented as a last-in-first-out (LIFO) memory. While it is not taught explicitly, it is inherent and is well known in the art that stacks are implemented as last-in first-out data structures.

13. Regarding claim 10, Patel has taught the apparatus according to claim 1, wherein said extension stack comprises both head (see "Optop", Col.4 lines 14-15) and tail interfaces (see "Frame", Col.4 lines 15-16).

14. Regarding claim 16, Patel has taught the apparatus according to claim 7, wherein said stack management unit (64 of Fig.3) is configured to control:

a. Pushes to said one or more internal registers from said extension stack (see Col.5 lines 6-11), and

b. Pops from said one or more internal registers to said extension stack (Col.5 lines 6-11).

15. Regarding claim 18, Patel has taught a method for implementing a Java virtual machine top of stack comprising the steps of:

a. Translating one or more instruction codes of a first instruction set into sequences of instruction codes of a second instruction set (see Col.3 lines 50-53 and Col.4 lines 1-4),

b. Manipulating contents of one or more internal registers (44 of Fig.3) of a processor in response to said sequence of instruction codes of said second instruction set (see Col.4 lines 6-22),

c. Implementing a stack comprising said one or more internal registers and an extension stack coupled between said processor and a memory device (50 of Fig.3), wherein said one or more internal registers (44 of Fig.3) are configured as a top of stack (see Col.4 lines 9-22). Here, the JVM stack is implemented using both the internal registers (44 of Fig.3) and the extension stack (50 of Fig.3), as

not only are the java registers in the internal registers stored within the extension stack (see Col.4 lines 19-22), but the internal registers contain the pointers to the extension stack, allowing data to be written to and read from the stack (see Col.4 lines 12-17).

16. Regarding claim 19, Patel has taught the method according to claim 18, wherein said instruction codes of said first instruction set comprise stack operations (see Col.2 lines 19-27). It is well known in the art that Java Virtual Machine instructions are stack-based, with their operands being provided on a stack rather than in registers. It is therefore inherent that any non-trivial instructions of the stack-based Java instruction set (first instruction set) will be "stack operations" due to their stack-based nature.

17. Regarding claim 23, Patel has taught the apparatus according to claim 1, further comprising:

a. A register block (44 of Fig.3) coupled between said processor interface and said extension stack and configured to operate as a bridge between said processor and said extension stack (see Fig.3).

Claim Rejections - 35 USC § 103

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

19. Claims 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al, U.S. Patent No. 6,332,215.

20. Regarding claim 14, Patel has taught the apparatus according to claim 7, but has not explicitly taught wherein said stack management unit is further configured to track which internal registers are used for the stack.

21. However, Patel has taught the tracking of another set of registers usage in a different stack (see Col.5 lines 43-47) so that it can be determined which operands are valid operands and thus which can be used in handling overflow/underflow (see Col.5 line 61 – Col.6 line 11). Because Patel has not taught explicit reasons why the same tracking of registers can't be used on the internal registers (44 of Fig.3) and the extension stack (50 of Fig.3) that are used to implement the stack, one of ordinary skill in the art would have found it obvious to modify Patel to further track which registers of the internal registers (44 of Fig.3) are used to implement the stack so that valid operands can be identified such that overflow/underflow situations can be handled correctly.

22. Regarding claim 15, Patel has taught the apparatus according to claim 14, wherein said stack management unit is further configured to track how many internal registers are used for the stack (see Patel, Col.4 lines 8-22 and Col.5 lines 34-37). Here, the operand stack is implemented using the Java registers (44 of Fig.3) and the java stack (50 of Fig.3), and counters keep track of how many entries have been placed on the operand stack (see Col.5 lines 34-37).

23. Claims 8, 11-13, 17 and 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al, U.S. Patent No. 6,332,215 as applied to claim 7 above, and further in view of Tremblay et al, U.S. Patent No. 6,021,469.

24. Regarding claim 8, Patel has taught the apparatus according to claim 7, wherein said stack management unit (64 of Fig.3) is configured to control transfers between:

a. Said internal registers and said extension stack (see Patel, Col.5 lines 6-11).

25. Patel has not explicitly taught wherein the stack management unit is configured to control transfers between the extension stack and said memory device.

26. However, Tremblay has taught the transferring of data from a stack to a data cache so that pipeline stalls due to overflow/underflow conditions can be avoided (see Tremblay, Col.18 lines 46-61). Therefore, one of ordinary skill in the art would have found it obvious to modify Patel such that the stack management unit could control transfers between the extension stack and a data cache so that costly pipeline stalls will be avoided.

27. Regarding claim 11, Patel in view of Tremblay has taught the apparatus according to claim 8, wherein said extension stack:

a. Is emptied to said memory device to prevent an overflow (see above paragraphs 26-28 and Tremblay, Col.18 lines 46-61), and

b. Filled from said memory device to prevent an underflow (see above paragraphs 26-28 and Tremblay, Col.18 lines 46-61).

28. Regarding claim 12, Patel in view of Tremblay has taught the apparatus according to claim 11, wherein said memory device comprises a main memory of said processor (see above paragraphs 26-29 and Tremblay, Col.18 lines 34-41).

29. Regarding claim 13, Patel has taught the apparatus according to claim 7, but has not explicitly taught wherein said extension stack is configured to indicate an almost empty or almost full condition.

30. However, Tremblay has taught the use of high and low watermarks to indicate when the stack is near full or empty so to avoid stalling the pipeline due to overflows or overwrites (see

Tremblay, Col.18 lines 46-61). Therefore, it would have been obvious to modify Patel to include high and low watermarks as Tremblay does to more clearly indicate when a potential overflow or overwrite condition occurs and avoid costly pipeline stalls.

31. Regarding claim 17, Patel has taught an apparatus comprising:

a. Means for manipulating data in response to instruction codes of a first instruction set (see Col.4 lines 6-22), said manipulating means comprising a number of internal registers (44 of Fig.3),

b. Means for translating instruction codes of a second instruction set into sequences of said instruction codes of a said first instruction set, wherein said translating means is configured to implement a stack (see Col.4 lines 9-22) with:

i. One or more of said internal registers (44 of Fig.3) and an extension stack coupled between said manipulating means and a memory device (50 of Fig.3). Here, the JVM stack is implemented using both the internal registers (44 of Fig.3) and the extension stack (50 of Fig.3), as not only are the java registers in the internal registers stored within the extension stack (see Col.4 lines 19-22), but the internal registers contain the pointers to the extension stack, allowing data to be written to and read from the stack (see Col.4 lines 12-17).

ii. Use said one ore more of said internal registers as a top of stack (see Col.4 lines 8-17),

iii. Transfer contents of said one or more internal registers to said extension stack (see Col.5 lines 6-11),

iv. Transfer contents of said extension stack to said one or more internal registers (see Col.5 lines 6-11).

32. Patel has not explicitly taught wherein the "stack" can empty to or refill from a memory device.

33. However, Tremblay has taught the emptying and refilling of data between a stack and a memory device so that pipeline stalls due to overflow/underflow conditions can be avoided (see Tremblay, Col.18 lines 46-61). Therefore, one of ordinary skill in the art would have found it obvious to modify Patel such that the implemented stack could empty to and refill from a memory device so that costly pipeline stalls will be avoided.

34. Regarding claim 20, Patel has taught the method according to claim 18, further comprising the step of:

a. Transferring values between said internal registers and said extension stack (see Col.5 lines 6-11) in response to a first one or more of said sequences of instruction codes of said second instruction set (see Col.4 lines 6-22),

35. Patel has not explicitly taught the transferring of values between the extension stack and said memory device in response to watermark indications from the extension stack.

36. However, Tremblay has taught the use of high and low watermarks to indicate when a stack is near full or empty so to avoid stalling the pipeline due to overflows or overwrites (see Tremblay, Col.18 lines 46-61). Therefore, it would have been obvious to modify Patel to include high and low watermarks and transfer values between the stack and a memory device as Tremblay does to more clearly indicate when a potential overflow or overwrite condition will occur and prevent it, thus avoiding costly pipeline stalls.

37. Regarding claim 21, Patel has taught the method according to claim 18, further comprising the step of:

a. Generating control signals configured to:

i. Transfer values from said one or more internal registers to said extension stack (see Col.5 lines 6-11),

ii. Restore values from said extension stack to said one or more internal registers (see Col.5 lines 6-11).

38. Patel has not explicitly taught the extension stack emptying to and refilling from said memory device in response to watermark levels.

39. However, Tremblay has taught the use of high and low watermarks to indicate when a stack is near full or empty so to avoid stalling the pipeline due to overflows or overwrites (see Tremblay, Col.18 lines 46-61). Therefore, it would have been obvious to modify Patel to include high and low watermarks and transfer values between the stack and a memory device as Tremblay does to more clearly indicate when a potential overflow or overwrite condition will occur and prevent it, thus avoiding costly pipeline stalls.

40. Regarding claim 22, Patel has taught the apparatus according to claim 1, wherein said translator circuit is configured to:

a. Transfer contents of said one or more internal registers to said extension stack (see Col.5 lines 6-11),

b. Transfer contents of said extension stack to said one or more internal registers (see Col.5 lines 6-11).

41. Patel has not explicitly taught the extension stack emptying to and refilling from said memory device in response to overflow or underflow conditions, nor has Patel taught the transferring of contents from one or more internal registers to or from the extension stack in response to overflow or underflow conditions.

42. However, Tremblay has taught the emptying and refilling of data between a stack and a memory device so that pipeline stalls due to overflow/underflow conditions can be avoided (see Tremblay, Col.18 lines 46-61). Therefore, one of ordinary skill in the art would have found it obvious to modify Patel such that the implemented stack could empty to and refill from a memory device so that costly pipeline stalls will be avoided.

43. Furthermore, Tremblay has taught the transferring of contents from one level of a memory hierarchy to another when there is an overflow/underflow condition (see Tremblay, Col.18 lines 27-45) in order to prevent pipeline stalls (see Tremblay, Col.18 lines 46-61). Because the internal registers and extension stack of Patel are a memory hierarchy (see Patel, Fig.3), one of ordinary skill in the art would have found it obvious to transfer contents of the internal registers to and from the external stack upon an overflow/underflow condition so that costly pipeline stalls are prevented.

Response to Arguments

44. Applicant's arguments filed 07 October 2004 have been fully considered but they are not persuasive. Applicant argues in essence on pages 10-12,

> "In contrast to Patel, the presently claimed invention (claim 1) provides an extension stack **coupled between a processor interface and a memory**

interface...Patel does not disclose or suggest each and every element of the presently claimed invention, arranged as in the present claims..."

45. This has not been found persuasive. The stack (Patel Figure 30, element 50) is found between the processor (Patel Figure 3, element 25) and the memory (Patel Figure 3, element 24). As shown in Patel's Figure 3, Patel's Java Stack 50 is connected to the Hardware Java Registers 44, which is connected to the Java Accelerator Instruction Translation 42. All of them are found between the memory, i.e. instruction cache 24, and the CPU 25, since they all receive the output from memory and input into the CPU. The Java Accelerator Instruction Translation 42 receives the memory output directly, the Hardware Java Registers 44 receives the output via the Java Translation 42, and the Java Stack received the output via the Registers 44. The Java Translation delivers its output via selector 28 to the CPU 25, the Registers 44 via the Registers 44, and the Java Stack 50 via the Registers 44. As seen in Patel's Figure 3, all connections between these three elements are two ways, meaning they all receive and deliver information and data to each other. Therefore, Patel's Java Accelerator Instruction Translation 42, Hardware Java Registers 44, and Java Stack 50 are all coupled between the memory (Instruction Cache 24) and CPU 25. These three elements are connected to the output of the memory interface and input to the CPU. The claim language merely states that the stack is "coupled between said processor interface and said memory interface", not that it must have direct connections to the processor interface and memory interface as the arguments seem to be suggesting. With the open claim language, the stack just needs to be found somehow connected between the processor interface and memory interface, i.e. the stack just needs to be found somewhere between the memory and processor and

can be connected to anything in between and this is shown in Patel's Figure 3. Therefore, the stack is coupled between the processor interface and memory interface.

Conclusion

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

47. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

48. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
7 January 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100